Specification

Problem Background

Wheelchair users currently board buses with the help of bus drivers. With driverless buses being deployed, such users may face difficulties getting aboard buses on their own.

Based on (Impact of Automation on Disabled People Summary Report), the problems faced by disabled individuals include:

- Being able to buy tickets, board and alight the vehicle, navigate to suitable seats, and the use of disabled priority spaces.
- **All stages of the journey need to be accessible for disabled people to be able to use automated transport**. It's not just about the vehicle itself, but the surrounding infrastructure. This includes **information provision, digital systems and interfaces,** transfer to other transport modes, and the built environment.
- A specific concern around automation was related to the lack of a driver, who would carry out an essential role in providing physical assistance, information, enforcement of rules and assistance in emergencies.
- **Developer priorities are typically ensuring that the technology functions correctly and safely to establish a solid foundation before focusing on the user experience**
- **Lack of information about the busyness of a service and how many accessible seats and wheelchair spaces there are.**
- **A key barrier 'pre-journey' is the need for reliable, accessible and up-to-date information regarding the service**. Rely on various sources of information (including tactile, audible, and visual messages).
- **Ensuring they know or can find out where you are and when your stop is**.
- **Individuals were not allowed on the bus and think this was because of something connected to their disability and had issues reporting such incidents.**
- **Individuals missed their stop due to lack of driver alerts**.

We will particularly handle the problems in bold.

The app will need to:

- Be predictable and reliable
- Provide effective assistance, available when needed, accessible information and communication
- Provide users with cues such as changes to their journey and upcoming stops

The bus model is the Alexander Dennis EV100, which comes with a manual ramp.

[Optionally] Provide app accessibility options for those with impairments to their hearing, vision and cognitive abilities e.g. text-to-speech, larger fonts etc.

Assumptions

Our app deals with 2 cases:

- Manual ramp with a safety driver
- Automated ramp without a safety driver

Disabled persons have access to a RFID/NFC card and the council is willing to install RFID/NFC scanners at bus stops.

The real-time bus data stores **bus capacity**, **priority seat capacity** as well **accessibility requirements**.

There is an API to communicate announcements or flag announcements on buses.

In the case that there is built-in functionality for reporting incidents:

- There is access to an API to communicate with the Council's database of reports/complaints.

Bus ramp has a weight sensor so it only retracts when no one has been on it for e.g. 5 seconds.

Functional Requirements

An app for users' mobile phones

- **Onboarding**: collect user data on accessibility requirements
- **Information about upcoming buses**: number of accessibility seats - also indicate if an occupied seat is expected to be vacated before reaching user, how busy the bus is, whether the ramp is manual/automatic and whether it's broken
- **Journey planning**: input a source and destination and plan a trip of accessible buses with timing estimations
- **Prime upcoming buses**: deploy ramp upon arrival, announcement to passengers to free disabled seats
- **Alerts**: Alert users when they have reached their destination
- **Feedback**: Contact to report instance of Equality Act breach (Possibly share contact points or build functionality in)
- **[optional] Chatbot**: To give advice and information
- ~~Notify buses to wait longer when disabled individuals board/exit bus~~

[Optional] The app could also work on tablets located at each bus stop with an NFC/RFID reader

System Components

Components of the system include:

- Server
- Bus stop app
- Client-side app
- Bus stop automated ramp API
- RFID reader at bus stop
- Safety drivers

Interface Specifications

Lo-fi prototype on Canva

RFID reader -> bus stop app: sends accessibility data
Bus stop app -> Council server: sends accessibility data
Client-side app -> Council server:

- Journey planner: sends request “for the next Bus A1, deploy a ramp when it reaches my bus stop” or “for the next Bus A1, the boarding passengers have the following requirements” - we just pass requirements

Council Server -> client-side app:

- Real-time bus location data

Buses -> Council server:

- Bus capacity
- How occupied the bus is
- If the wheelchair spaces are occupied
- Position
- Delay/expected arrival time
- Request acknowledgement

Acceptance Criteria

A working Minimum Viable Product that:

- Satisfies the mandatory functional requirements
- Is in line with PSVAR, the Equality Act and GDPR data protection regulation.

Management Strategy

- Communication to client via email
- Communication among group members on WhatsApp
- Possibly weekly/fortnightly meetings
- Code in GitHub repository, monitor individual progress via commit log

Participatory user-centric design

As stated, in Impact of Automation on Disabled People Summer Report, one of the drawbacks in most projects concerning accessibility for those impaired, it was voiced that there was a lack of involvement of affected parties in solutions leading to misalignment and barriers.

We aim to solve this by actively involving disabled individuals in our ideation, design and implementation process by reaching out to:

- Transport For All
-

Technical Contributions by each member

1. Frontend UI/UX (2 people)
2. Backend - Fetching real-time bus location data and specifications and pushing to and from the Council server (2 people)
3. Networking - communication between app and in-house server (1 person)
4. [Optional] Handling RFID/NFC scanner communication with the backend and card encoding